Inference in particle tracking experiments by passing messages between images

M. Chertkov [1,2], L. Kroc [1], F. Krzakala [1,3], M. Vergassola [4], and L. Zdeborová [1]

[1] *Theoretical Division and Center for Nonlinear Studies,*
Los Alamos National Laboratory, NM 87545 USA [2] *New Mexico Consortium,*
Los Alamos, NM 87544 [3] *ESPCI ParisTech and UMR 7083 "Gulliver",*
France [4] *Institut Pasteur; CNRS URA 2171, F-75015 Paris, France*

Methods to extract information from the tracking of mobile objects/particles have broad interest in biological and physical sciences. Techniques based on simple criteria of proximity in time-consecutive snapshots are useful to identify the trajectories of the particles. However, they become problematic as the motility and/or the density of the particles increases due to uncertainties on the trajectories that particles followed during the images' acquisition time. Here, we report an efficient method for learning parameters of the dynamics of the particles from their positions in time-consecutive images. Our algorithm belongs to the class of message-passing algorithms, known in computer science, information theory and statistical physics as Belief Propagation (BP). The algorithm is distributed, thus allowing parallel implementation suitable for computations on multiple machines without significant inter-machine overhead. We test our method on the model example of particle tracking in turbulent flows, which is particularly challenging due to the strong transport that those flows produce. Our numerical experiments show that the BP algorithm compares in quality with exact Markov Chain Monte-Carlo algorithms, yet BP is far superior in speed. We also suggest and analyze a random-distance model that provides theoretical justification for BP accuracy. Methods developed here systematically formulate the problem of particle tracking and provide fast and reliable tools for its extensive range of applications.

Tracking of mobile objects is widespread in the natural sciences, with numerous applications both for living and inert "particles". Trajectories of the particles are to be obtained from successive images, acquired sequentially in time at a suitable rate. Examples of living "particles" include birds in flocks [1] and motile cells [3]. Among inert objects, nanoparticles [4] and particles advected by turbulent fluid flow [5–7] provide two important examples. The general goal of tracking particles is to extract clues about their dynamics and to make inferences about the laws of motion and/or unknown modeling parameters.

Ideal cases for tracking are those where the density and the mobility of particles is low and the acquisition rate of images is high. The non-dimensional parameter governing the stiffness of the problem is the ratio $\Lambda = \ell \rho^{1/d}$ of the typical distance ℓ traveled by the particles during the time between images and the average inter-particle distance $1/\rho^{1/d}$. Here, ρ is the number density of particles and d is the space dimensionality. Tracking is rather straightforward if Λ is small: the positions of each particle in two successive images will be relatively far from those of all other particles. Trajectories are thus defined without ambiguity. Such a situation is encountered for instances of the tracking of nanoparticles [8]. More generally, effective methods are available to identify the assignment (defined as a one-to-one mapping of the particles from one image to the next one, i.e., the set of trajectories for all tracked particles) that is the most probable [9–11].

The level of difficulty soars as Λ increases: many sets of trajectories, i.e., many mappings among particles in successive images, have comparable likelihoods, see Fig. 1. The dynamics of the particles ought to be described by an explicit model, which generally features unknown parameters. The model defines a probability distribution over the space of all possible assignments. Contrary to the small Λ case, the probability distribution is not necessarily dominated by a unique assignment. Notwithstanding this uncertainty on the trajectories, it is expected that useful information might still be extracted if the number of tracked particles is sufficiently large. The difficulty is that all possible assignments must be considered: restricting to the most probable assignment (MPA) generally leads to biased inferences (see the sequel). Reliable inference requires summing over all possible assignments with their appropriate weights. Problems with large Λ occur in practice; e.g., even with state-of-the-art cameras, particles in turbulent flow [12] and birds in flocks [1] often feature ambiguities in the reconstruction of particles' trajectories. Developing systematic methods to tackle these cases constitutes our scope here.

The plan of the paper is as follows. First, we formulate the problem of particle tracking in terms of a graphical model. We then show that an exact and rapid algorithm for summing over all possible assignments is unlikely to become available, as such an algorithm could equivalently compute the permanent of a non-negative matrix, a problem that is well-known to be #P-complete [13]. An approximate message-passing Belief Propagation (BP) algorithm [14–16] is then introduced, employed and tested. We also introduce a simplified model where analytical results and a quantitative sense of the BP approximations are obtained. The Results section presents numerical simulations comparing results of BP and the inference based on the MPA. For the sake of concreteness, we consider the case of particles passively transported by a turbulent flow, but the methods are quite general and can be applied to other
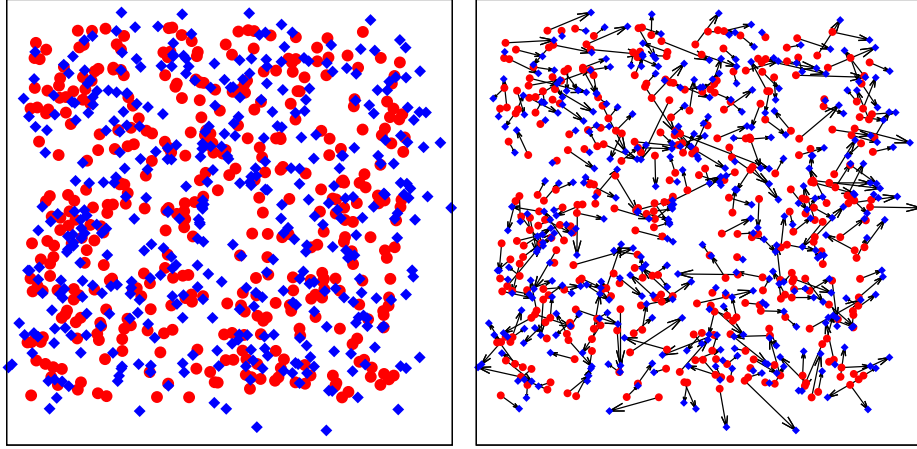
FIG. 1: A concrete example of particle tracking, with $N = 400$ particles moving from their original positions (red circles) to new ones (blue diamonds). In the left figure, two consecutive images are superimposed to facilitate comparison of the successive positions. Particles are transported by a turbulent fluid flow with local stretching, shear, vorticity and diffusivity parameters $a^* = 0.28$, $b^* = 0.54$, $c^* = 0.24$ and $\kappa^* = 1.05$ (see eq. (9)). We focus on turbulent transport because of the challenges it poses, yet the methods we develop are quite general. The right figure shows the actual motion of each particle. Evidently, the simple criterion of particle proximity fails to pick the actual trajectories, and the mapping of the particles between the two images is intrinsically uncertain. Nevertheless, the inference algorithms described here rapidly yield excellent predictions ($a = 0.32$, $b = 0.55$, $c = 0.19$ and $\kappa = 1.00$) for the parameters of the flow.

situations as well.

I. MODELS

Tracking of particles as a graphical model. Graphical models provide a framework for inference and learning problems widespread in machine learning, bioinformatics, statistical physics, combinatorial optimization and error-correction [14, 15, 20]. The assignment problem involved in the tracking of particles is conveniently recast as a weighted complete bipartite graph (see Fig. 2). Nodes are associated with the N particles in each of two successive images, their positions being denoted $\boldsymbol{x}_i$ and $\boldsymbol{y}^j$, respectively. These $2N$ position vectors ($i, j = 1, \ldots, N$) constitute the experimental data provided. We suppose that a model for the dynamics of the particles is available and features a set of unknown parameters $\boldsymbol{\theta}$. Edges between nodes of the bipartite graph are weighted according to the likelihood that, according to this model, a particle moves from the initial position $\boldsymbol{x}_i$ to the final position $\boldsymbol{y}^j$. Specifically, the formula for the likelihood of an assignment among (non-interacting) particles in two images reads as follows:

$$\mathcal{L}(\{\sigma\}, \boldsymbol{\theta}) = C\left(\{\sigma\}\right) \prod_{(i,j)} \left[P_i^j \left(\boldsymbol{x}_i, \boldsymbol{y}^j | \boldsymbol{\theta} \right) \right]^{\sigma_i^j} . \tag{1}$$

The Boolean variable σ_i^j indicates whether the particles i and j are matched ($\sigma_i^j = 1$) or not ($\sigma_i^j = 0$). The set of the N^2 variables σ_i^j is denoted by $\{\sigma\}$. The constraint function $C\left(\{\sigma\}\right) \equiv \prod_j \delta \left(\sum_i \sigma_i^j, 1 \right) \prod_i \delta \left(\sum_j \sigma_i^j, 1 \right)$, involving Kronecker δ functions, enforces the conditions for a perfect matching, i.e., a one-to-one correspondence between the particles in the two images. (Situations where the number of particles in the two images can differ and/or positions of the particles are uncertain are accommodated within the same formalism presented in the sequel; see Supplementary Information (SI).) The quantity P_i^j is the transition probability that a particle at position $\boldsymbol{x}_i$ travels to $\boldsymbol{y}_j$ in the time Δ between images. The transition probabilities carry all of the information about the model for the dynamics of the particles.

The tracking inference problem that we address here is to provide fast and reliable estimates of the model's unknown parameters $\boldsymbol{\theta}$, which enter the likelihood of the trajectories via eq. (1).

The simplest method to infer the unknown parameters $\boldsymbol{\theta}$ is first to identify for any $\boldsymbol{\theta}$ the MPA, i.e., a configuration $\{\sigma\}$ satisfying the constraint $C(\{\sigma\}) = 1$ and having the highest likelihood (1), and then to maximize the resulting
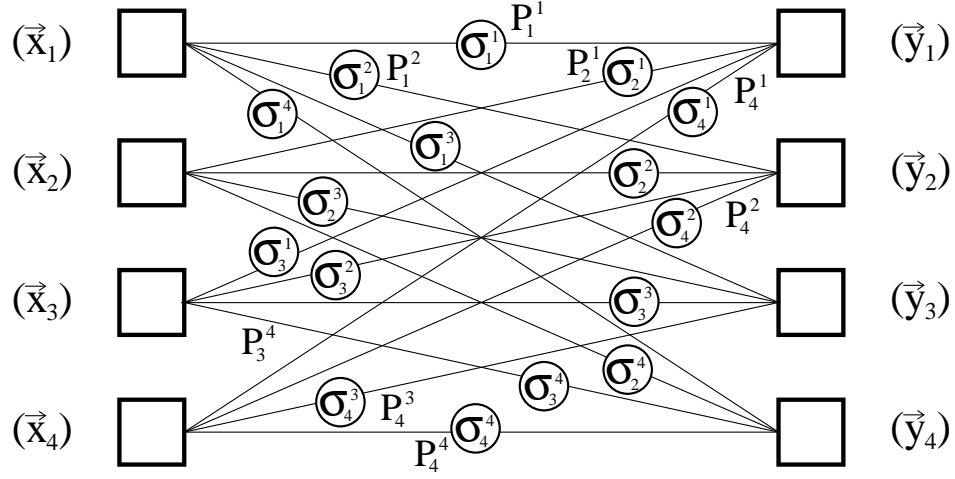
FIG. 2: The complete bipartite graph for the tracking problem. Nodes (squares) denote particles in two consecutive images. Edges carry weights P_i^j (the likelihood that a particle travels from the initial position $\boldsymbol{x}_i$ to the final position $\boldsymbol{y}^j$) and Boolean variables σ_i^j (indicating whether the nodes i and j in the two images correspond to each other ($\sigma = 1$) or not ($\sigma = 0$)). Conflicts arise from the constraint that a valid assignment is a one-to-one mapping of the particles in the two images (expressed by the constraint $C(\{\sigma\})$ in eq. (1)).

likelihood with respect to $\boldsymbol{\theta}$. Exact polynomial algorithms [9, 10] solve the first MPA task. The surprising remark recently made in [11, 18] is that the auction exact algorithm can be reformulated as a message-passing Belief Propagation (BP) scheme. (Surprise stems from the fact that BP usually differ from the exact solution of a problem if loops are present in its graph.)

It is however expected—and confirmed by results described shortly—that the MPA provides reliable inferences only for small enough values of the stiffness parameter Λ defined in the Introduction. As Λ increases, assignment-dependent entropic factors become important, and MPA inferences deviate from the actual values of the parameters $\boldsymbol{\theta}$. This deviation is understandable because the Bayesian probability distribution for the parameters, assuming uniform prior probability for the assignments, involves the full likelihood (1) marginalized over all possible assignments:

$$P(\boldsymbol{\theta}|\boldsymbol{x}_i, \boldsymbol{y}^j) \propto \sum_{\{\sigma\}} \mathcal{L}\left(\{\sigma\}|\boldsymbol{\theta}\right) \equiv Z\left(\boldsymbol{\theta}\right) . \tag{2}$$

As the stiffness parameter Λ increases, many assignments become likely and the sum (2) is not dominated by the MPA.

Summing over all possible trajectories. In the vast majority of cases, no exact algorithm is available to sum over all possible states of a graphical model. Equation (2) is no exception, as it can also be seen as a sum over all possible permutations of the lower indices i into the upper indices j. It is then recognized that computing the likelihood Z (also known as partition function in statistical physics) is equivalent to computing the permanent of the matrix P_i^j, a well-known #P-complete problem [13]. However, the matrix P_i^j in our sum (2) is non-negative and the permanent of non-negative matrices was the first #P-complete problem discovered to be solvable by a Fully Polynomial Randomized Approximation Scheme (FPRAS) [17]. The complexity of the original FPRAS algorithm is $O(N^{11})$. We significantly accelerated (to $O(N^3)$) and simplified the original Markov Chain Monte-Carlo (MCMC) algorithm of [17] without observable deterioration of its quality (see SI). In the Results section, we use this simplified version to assess the accuracy of our BP approximation while in the SI we compare performance of BP to the MCMC scheme.

Among the possible approximations to compute the permanent of a matrix, Belief Propagation (BP) has a special status because of its aforementioned exactness for the MPA problem [11]. Moreover the BP algorithm is very fast, scaling as $O(N^2)$ in its basic form and linearly if, for each particle, only a limited number of nearby particles is considered. We shall therefore pursue the development of BP to approximate the sum in (2) and then assess its validity through numerical simulations and the simplified "random distance" model discussed shortly.

The starting point of the BP approach is the remark that the convex Kullback-Leibler functional

$$\mathcal{F}\{b(\{\sigma\})\} \equiv \sum_{\{\sigma\}} b(\{\sigma\}) \ln \frac{b(\{\sigma\})}{\mathcal{L}(\{\sigma\})} , \tag{3}$$

has a unique minimum at $b(\{\sigma\}) = \mathcal{L}(\{\sigma\})/Z$ (under the normalization condition $\sum_{\{\sigma\}} b(\{\sigma\}) = 1$), where $\mathcal{L}$ is defined in eq. (1) and Z in eq. (2). The corresponding value of the functional $\mathcal{F}$ is the log-likelihood (free energy), $\mathcal{F} = -\ln Z$. This remark constitutes the basis for variational methods (see [15]), where the minimum of the functional is sought in a restricted class of functions. BP (and the corresponding approximation for the free energy, named in [16] after Hans Bethe) involves an ansatz of the form

$$b(\{\sigma\}) \approx \frac{\prod_i b_i(\boldsymbol{\sigma}_i) \prod_j b^j(\boldsymbol{\sigma}^j)}{\prod_{(i,j)} b_i^j(\sigma_i^j)} \,, \tag{4}$$

where each vector $\boldsymbol{\sigma}_i \equiv \{\sigma_i^j | j = 1, \cdots, N\}$ can be any of the N possible vectors $(0, \cdots, 0, 1, 0, \cdots, 0)$ having exactly one nonzero entry, and $\boldsymbol{\sigma}^j$ is defined analogously. This ansatz is motivated by the fact that the probability distribution on a graph with a tree structure, i.e., without loops, takes exactly the form (4). The quantities b_i, b^j, and b_i^j are called "beliefs". In the absence of loops, b_i^j represents the probability distribution for the single Boolean variable σ_i^j, and b_i represents the joint probability distribution of the components of $\boldsymbol{\sigma}_i$, which appear in a constraint. Without loops, the beliefs automatically satisfy the conditions of marginalization, viz., $b_i^j(\sigma_i^j) = \sum_{\boldsymbol{\sigma}_i \backslash \sigma_i^j} b_i(\boldsymbol{\sigma}_i)$, where the sum is performed over all possible values of the vector $\boldsymbol{\sigma}_i$ having the specified value of its component σ_i^j.

In the presence of loops, the expression (4) is no longer exact, and the marginalization conditions must be imposed as additional constraints. The important point demonstrated in [16] is that minimizing the functional (3) for the class of functions (4) (under the marginalization and the normalization conditions) yields the message-passing formulation of BP, which was heuristically introduced by Gallager for decoding of sparse codes [19].

Most commonly, the BP equations are formulated as an iterative message-passing scheme [15, 20], where the message $\underline{h}^{i \to j}$ (respectively, $\overline{h}^{i \to j}$) is sent from from particle i in the first (second) image to particle j in the second (first) image. In the case of the matching problem, the messages are determined by solving the following equations:

$$\overline{h}^{i \to j} = -\frac{1}{\beta} \ln \sum_{k \neq j} P_i^k e^{\beta \underline{h}^{k \to i}} \,\, ; \,\, \underline{h}^{j \to i} = -\frac{1}{\beta} \ln \sum_{k \neq i} P_k^j e^{\beta \overline{h}^{k \to j}} \,. \tag{5}$$

The "inverse temperature" β can be set to unity, but it is usefully retained to show that the limit $\beta \to \infty$ yields the exact solution for the MPA problem [11].

To solve the BP equations (5), the messages $(\{\overline{h}\}, \{\underline{h}\})$ are randomly initialized and iteratively updated in order to find a fixed point of the message-passing equations (5). The Bethe free energy then reads

$$-\beta \mathcal{F}_{BP}\left(\{\overline{h}\}, \{\underline{h}\}, \boldsymbol{\theta}\right) = \sum_{(ij)} \ln \left(1 + P_i^j e^{\beta \overline{h}^{i \to j} + \beta \underline{h}^{j \to i}}\right) -$$

$$\sum_i \ln \left(\sum_j P_i^j e^{\beta \underline{h}^{j \to i}}\right) - \sum_j \ln \left(\sum_i P_i^j e^{\beta \overline{h}^{i \to j}}\right) \,. \tag{6}$$

The Bethe free energy $\mathcal{F}_{BP}$, evaluated at the fixed point of the BP equations (5), provides an estimate of the exact log-likelihood $-\ln Z(\boldsymbol{\theta})$ defined through (2). Because the most likely set of parameters $\boldsymbol{\theta}$ minimizes $-\ln Z(\boldsymbol{\theta})$, we seek parameters that minimize the estimated free energy. We perform this minimization using Newton's method in combination with message-passing: after each Newton step, we update the messages $(\{\overline{h}\}, \{\underline{h}\})$ according to (5) and the current set of parameters $\boldsymbol{\theta}$. As this combined update takes N^2 steps and the number of combined step which led to convergence is N-independent, the running time scales as $O(N^2)$. Note that the running time can be further reduced to $O(N)$ neglecting the contribution of edges with very small probability P_i^j, i.e. diluting the fully connected bipartite graph.

A simplified model to understand the BP approximation. The BP approximation is exact only if the underlying graph is a tree, which is not the case for our fully connected bipartite graph in Fig. 2. In the Results, we empirically assess the validity of the BP approximation through numerical simulations. To supplement the numerical evidence, we introduce here a simplified "random distance" model for which analytical results can be obtained and used to understand the nature of the BP approximation.

The "random distance" model is defined as follows. First, we decouple the N^2 distances d_i^j between particles i and j by assuming that they are independent among each other. We then assume that one permutation π^* of the lower indices i into the upper indices j has a special status, while all other distances d_i^j are drawn independently at random from a given distribution. Namely, the N distances $d_i^{j=\pi_i^*}$'s are distributed as a Gaussian (restricted to positive values) with variance $\kappa^* = O(1)$. For each of the other $N(N-1)$ pairs (i,j), the distances d_i^j are independent
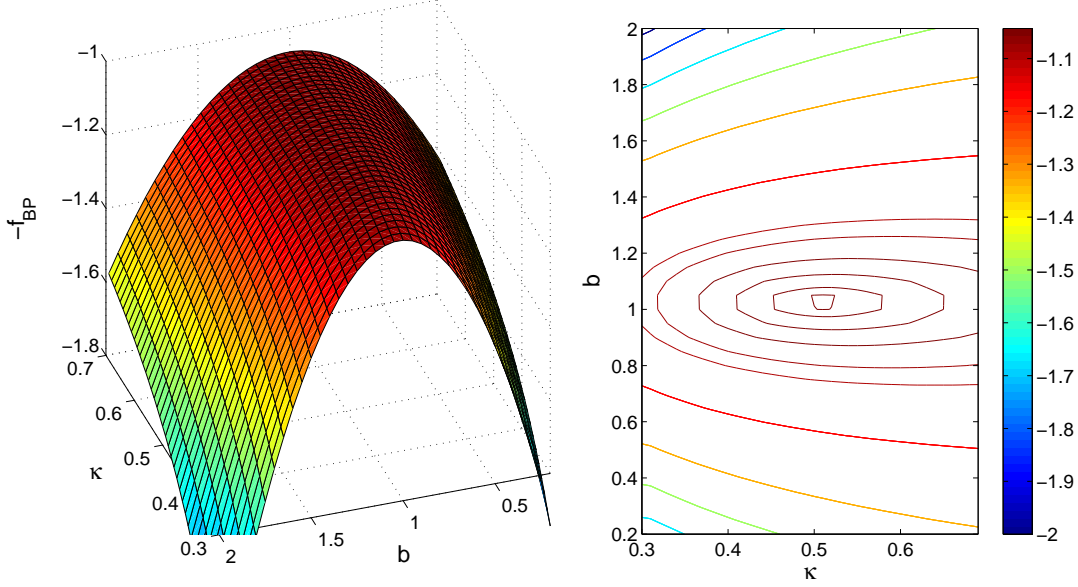
FIG. 3: A realization of a 2-dimensional flow with $a^* = b^* = c^* = 1$, $\kappa^* = 0.5$ and $N = 200$ particles. Left: The BP Bethe free energy as a function of the diffusivity κ and the shear b, where every point is obtained by minimizing with respect to the stretching a and the vorticity c of the flow. Right: The same free energy in a contour plot, showing the maximum close to $b = 1$ and $\kappa = 0.5$. The maximum is achieved for $a_{BP} = 1.148(1)$ $b_{BP} = 1.026(1)$ $c_{BP} = 0.945(1)$, $\kappa_{BP} = 0.509(1)$, where the parentheses indicates numerical error on the third digit.

random variables drawn uniformly in the interval $(0, N)$. Units of length are chosen so that the typical interparticle distance is set to unity. Note that any distribution of d_i^j for $j \neq \pi_i^*$ with a vanishing derivative at the origin would give the same solution. Indeed, the crucial property is that, for each particle i, the number of distances d_i^j that are comparable with the diffusion length scale $\sqrt{\kappa^*}$ is $O(1)$.

The interest of a special permutation π^* is that we can inquire about : learning κ^* if π^* is supposed unknown; the relevance of entropic factors for the partition function; and the status of the BP approximation. The same questions arise for (2) in the original problem. Note that if there were no special permutation, then we would obtain the random-link model considered in [22–24]. Our "random distance" model can be solved exactly in the thermodynamic limit using the replica method, as in [22, 23], or using the cavity method, as in [24]. The main result (see SI) is that the BP expression of the free energy for the "random distance" model is exact in the limit $N \to \infty$, despite of the short loops in the graph of the model. The argument to prove this result goes as follows : (a) the contribution to the partition function (2) from those permutations of the j indices that contain distances larger than $O(1)$ is negligible; (b) as each node has only a fraction $O(1)$ of its N distances being $O(1)$, the underlying graph is effectively sparse; (c) because sparse graphs are locally tree-like and correlations in the matching problem decay very fast on trees, it follows that the BP approximation is exact in the thermodynamic limit. We have formalized these statements within the replica and cavity methods (and rigorous local weak convergence methods are probably also applicable, as for the random link model [25]).

The asymptotic exactness found in the "random distance" model means that errors made by BP are caused by correlations among the inter-particle distances. In a smooth flow (see the sequel) particles close to each other in the first image will also be near each other in the second image; moreover, the four distances among the two positions in the first image and the two positions in the second will also be small, or more generally correlated. This effect is more important in lower dimensions. Indeed, BP inferences turn out to be better in the three-dimensional (3D) case than in 2D and rather inaccurate in 1D (with maximum relative error about 60%).

Our replica calculations also show that the "random distance" model presents an interesting phase transition at the diffusivity $\kappa_c \approx 0.174$. For $\kappa^* < \kappa_c$, the MPA π_{MPA} is identical to the special one π^* with high probability, whereas for $\kappa^* > \kappa_c$ the overlap (defined via the Hamming distance) between the most likely assignment π_{MPA} and the special one π^* is extensive, i.e., $O(N)$. The comparison with the finite-dimensional case is discussed in the Results section.
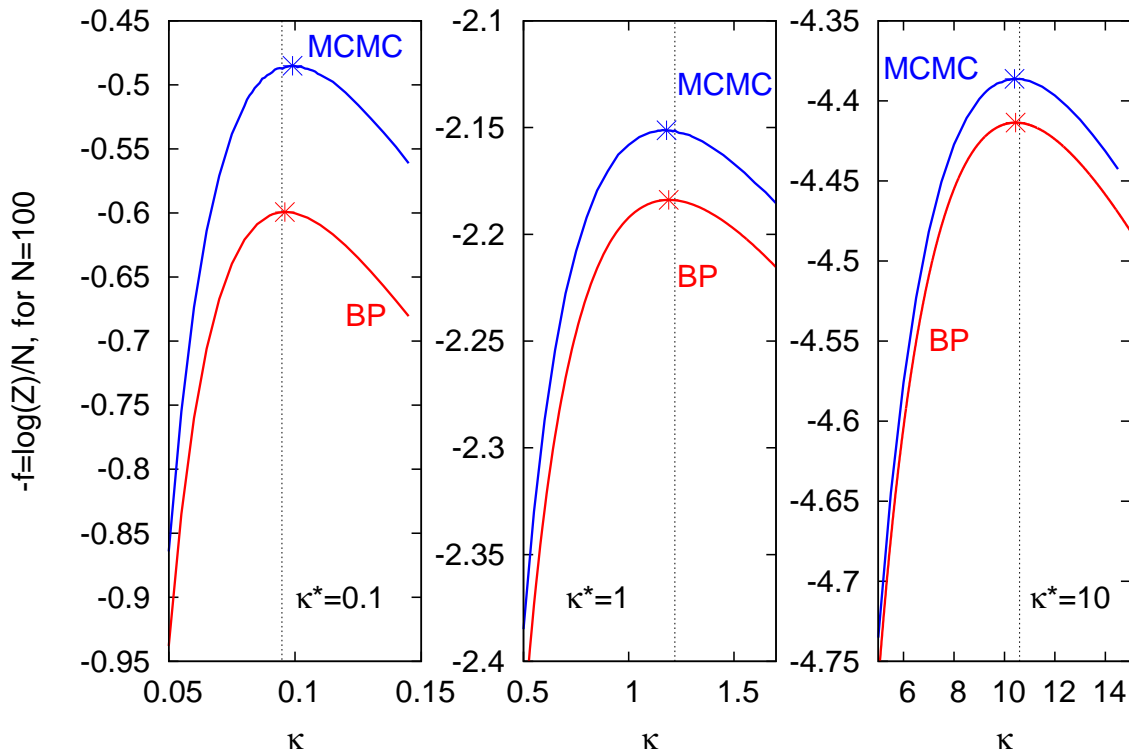
FIG. 4: The BP estimate of the log-likelihood $-\mathcal{F}_{BP}$ and the Monte-Carlo Markov-Chain estimate plotted *vs.* the diffusivity κ with $\kappa* = 0.1, 1, 10$ for $N = 100$ particles diffusing in 3D. Although the BP log-likelihood is significantly lower than the MCMC one, the estimate of the diffusivity is extremely good. The vertical lines mark the diffusivity computed based on the knowledge of the actual displacements. The BP algorithm compares favorably in estimation of the maximum with the basically exact Markov Chain Monte-Carlo algorithm, while being far superior in speed (for details on speed comparison see SI).

II. RESULTS

Our analysis has been quite general so far. To concretely assess the validity of BP, it is now necessary to specify the model appearing in the likelihood (1), namely the probability P_i^j for the transition from position $\boldsymbol{x}_i$ to $\boldsymbol{y}^j$ in the image acquisition time Δ. We decided to focus on the tracking of particles in turbulent flow for three reasons. First, the problem is highly relevant as an important part of modern experiments in fluid dynamics is based on the tracking of particles either in simple [5–7] or complex [26] flows. Second, all the algorithms used so far for reconstructing the flow from images of multiple particles have ignored the probabilistic structure of the possible assignments. As discussed in the review [12], the general approach is to search for a single matching. Criteria based on proximity and/or minimal acceleration identify for each particle its "best" mapping in the successive time-image. Conflicts where two or more particles in the first image are assigned to the same particle in the second image are resolved by various heuristics. The simplest option is to give up on those situations where a conflict arises; the most elaborate solution is to compute the assignment with the minimal cost in terms of proximity or acceleration. The bottomline is that one is always left with a single assignment, which leads to predictions that are effective at low density of the particles but rapidly degrade as their density increases [12]. Third, the laws of motion of the particles are well-known. Indeed, if particles are sufficiently small and chosen of appropriate (mass) density, their effect on the flow is negligible and they are transported almost passively. The (number) density of particles is usually rather high and a single snapshot contains a large number thereof. The reason is that the smallest scales of the flow ought to be resolved. Furthermore, turbulence is quite effective in rapidly transporting particles so that the acquisition time between consecutive snapshots should be kept small. Modern cameras have impressive resolutions, on the order of tens of thousands frames per second. The flow of information is huge: $\sim$ Gigabit$/s$ to monitor a two-dimensional slice of a $(10cm)^3$ experimental cell with a pixel size of $0.1mm$ and exposure time of $1ms$. This high rate makes it impossible to process data on the fly unless efficient algorithms are developed.

Tracking of particles in a turbulent flow. The likelihood P_i^j in eq. (1) for the transport of particles in turbulent flow is obtained as follows. Tracked particles are supposed to be at reciprocal distances smaller than the viscous scale of
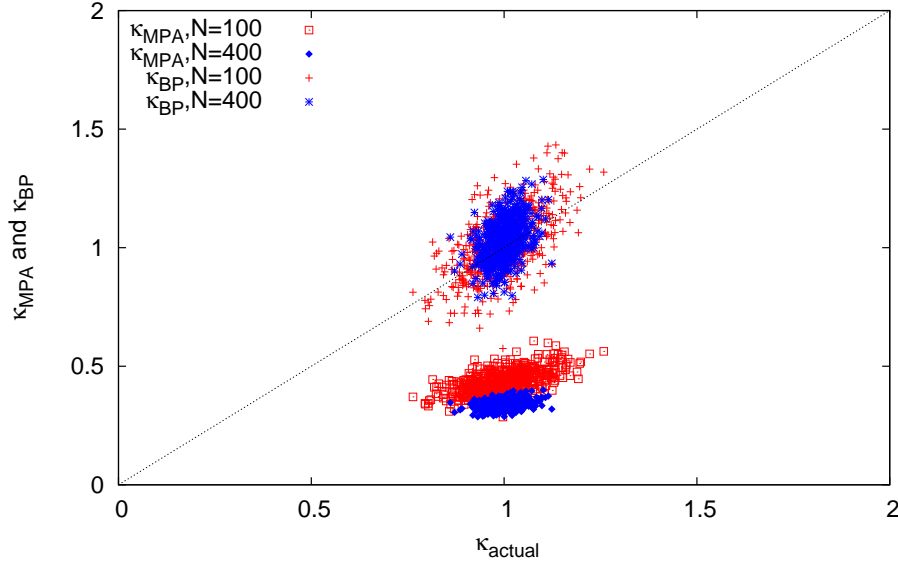
FIG. 5: Scatter plot of the diffusivity estimated by BP and by MPA *vs.* the actual value of the diffusivity. Diffusion takes place in 3D, with displacements generated using $\kappa^* = 1$. The actual value of the diffusivity κ_{actual} is computed from the actual trajectories and is subject to statistical fluctuations. The number of tracked particles is $N = 100$ (red) and $N = 400$ (blue) using 1000 measurements. The BP predictions correspond to the maximum of the log-likelihood, as approximated by the Bethe free energy (6) discussed in the text. Note the strong underestimation of the MPA estimate, to be contrasted with the cloud of BP predictions centered around the correct value of κ^*.

the flow. It follows (see [21] for a review) that the position $\boldsymbol{r}_i(t)$ of the i-th particle evolves according to the Lagrangian stochastic equations: $\dot{\boldsymbol{r}}_i = \hat{s} \cdot \boldsymbol{r}_i + \boldsymbol{\xi}_i$. Positions are measured with respect to a reference point and $\hat{s}$ is the tensor of the velocity derivatives. In two-dimensional incompressible flow $a = s_{xx} = -s_{yy}$ is the rate of stretching, $b = (s_{xy} + s_{yx})/2$ is the shear, and $c = (s_{xy} - s_{yx})/2$ is the vorticity. The stochastic term $\boldsymbol{\xi}_i(t)$ is the zero-mean Gaussian Langevin noise, describing molecular diffusivity, defined by its correlation function: $\langle (\xi_i)_\alpha(t_1)(\xi_j)_\beta(t_2) \rangle = 2\kappa \delta_{ij} \delta_{\alpha\beta} \delta(t_1 - t_2)$. The Greek indices refer to space components. The transition probability corresponding to the previous transport process is Gaussian:

$$P_i^j(\boldsymbol{x}_i, \boldsymbol{y}^j) = (\det M)^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2} r^\alpha (M^{-1})^{\alpha\beta} r^\beta\right); \tag{7}$$

$$\boldsymbol{r} = \boldsymbol{y}^j - W(\Delta)\,\boldsymbol{x}_i; \tag{8}$$

$$M = \kappa\, W(\Delta) \left[\int_0^\Delta W^{-1}(t) W^{-1,T}(t)\, \mathrm{d}t \right] W^T(\Delta), \tag{9}$$

where $W(t) = \exp(t\,\hat{s})$ [35] and Δ denotes the image acquisition time. The general problem of estimating the unknown parameters $\boldsymbol{\theta}$ now takes the special form of inferring the components of the tensor $\hat{s}$ and the diffusivity κ.

We tested the BP inference method on numerically simulated data. To compare different system sizes, we placed the N particles at random in a d-dimensional box of size $L = N^{1/d}$, i.e., the average density equals unity. Particles are then displaced independently following the probabilistic distribution (7) with a set of parameters $a^* N^{-1/d}$, $b^* N^{-1/d}$, $c^* N^{-1/d}$, κ^*. The timescale was chosen as the acquisition time $\Delta = 1$ and the parameters a, b, c of the flow were rescaled by $N^{-1/d}$, so that the particle displacements in the acquisition time are $O(1)$ for all choices of N. Fig. 3 shows the BP free energy (6) as a function of the shear b and the diffusivity κ for $N = 200$ particles. The curvature around the minimum of the exact free energy is inversely related to the statistical error in the estimation of the parameters. Figure 3 clearly shows that the most problematic parameter is the diffusivity κ, as confirmed by all of the numerical simulations we performed.

In the next three figures, we concentrate on the purely diffusive regime (where $a^* = b^* = c^* = 0$); subsequently, we return to the general case. Note that light scattering experiments provide an established measurement method for biological systems [27–30]. The technique is not commonly used in fluid dynamics experiments because the dispersion of the particles is much faster and the typical illumination level is too weak.

In Fig. 4 we compare the BP results with results from the fully polynomial randomized approximation scheme based on the Monte Carlo Markov Chain (MCMC) method. As computed by MCMC (which is guaranteed to be a FPRAS), the maximum of $Z(\kappa)$ coincides with the true value κ^*. From the large deviation theory interpretation of

$\mathcal{F}$, it follows that the statistical error of the estimation of κ is $\sigma = 1/[\sqrt{N\mathcal{F}''(\kappa)}]$. Figure 4 confirms the expectation that the curvature at the minimum of the free energy decreases with the diffusivity constant κ.

In Fig. 5 we compare the estimates of the diffusivity using BP (κ_{BP}) and using MPA (κ_{MPA}). The actual value κ_{actual} (respectively, the MPA estimate κ_{MPA}) is computed as the mean-square displacement of the particles on the actual (resp., the most probable) trajectories of the particles. The key conclusion to be drawn from Fig. 5 is that MPA largely underestimates the diffusivity, whereas the BP method is accurate.

Fig. 6 gives a quantitative sense of the BP accuracy as a

function of the diffusivity. The upshot of the curves is that MPA gives accurate estimates only at extremely low diffusivities. Conversely, as the diffusivity increases and the overlap among possible assignments becomes important, the quality of MPA predictions degrades very rapidly. Results from a similar study for 2D flows is presented in Fig. 7; vectorial parameters are again found to be computed efficiently by the BP method.
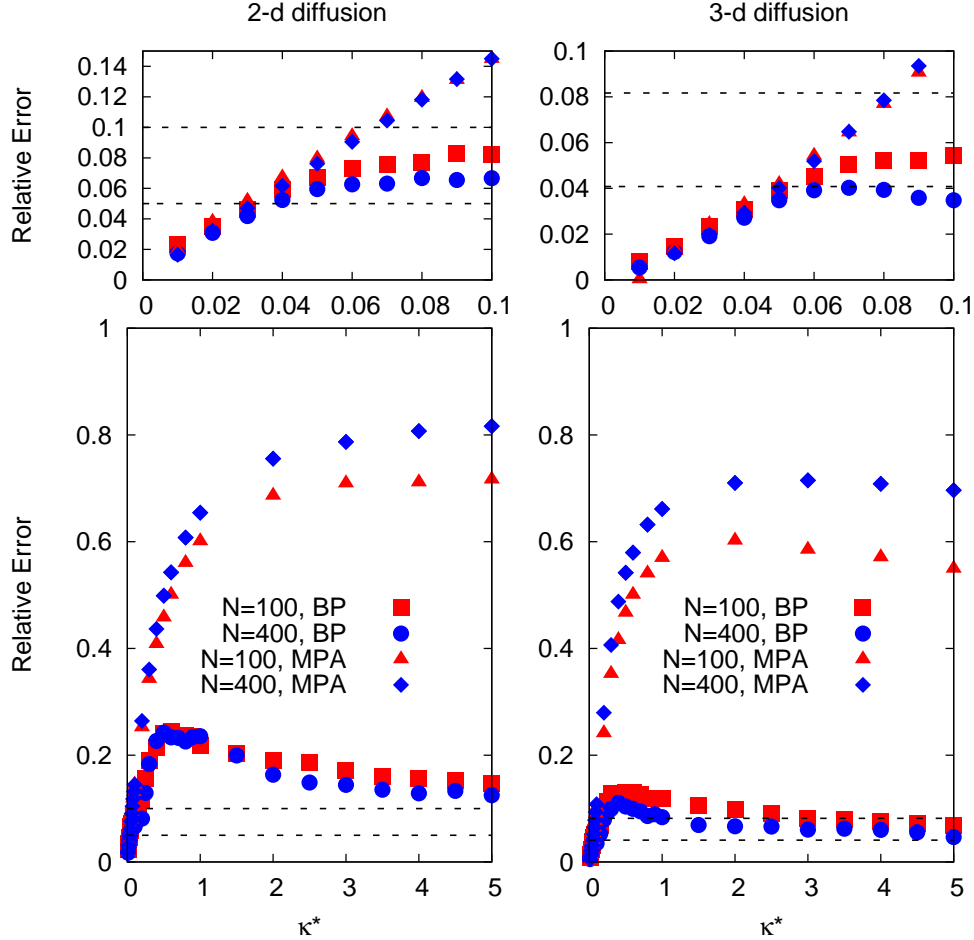

FIG. 6: The relative error $\sqrt{\sum(\kappa_{BP/MPA} - \kappa_{\text{actual}})^2/K}/\kappa_{\text{actual}}$ in the estimates of the diffusivity over K measurements *vs.* the actual value of the diffusivity κ^*. Circles and squares refer to Belief Propagation (BP) while triangles and lozenges refer to the Most Probable Assignment (MPA). κ_{actual} is the actual value of the mean-square displacement of the particles, i.e., it includes fluctuations around κ^* due to the finite number N of particles. The data are averaged over 1000 (for $N = 100$) and 250 (for $N = 400$) realizations and compared to relative the statistical error $\sqrt{2/dN}$ (dashed horizontal lines). The case of two-dimensional (2D) diffusion is shown on the left side, and the 3D case is on the right side. The top figures are zooms into the low diffusivity region.

Finally, Fig. 6 (top panel) indicates that the phase transition in the exactness of the optimal assignment, which was previously found for our "random distance" model, is smeared out in the finite dimensional case (or it happens only at $\kappa < 0.01$). This effect is traced to the fact that, in finite dimension, there are always several pairs of particles at distance $o(1)$ that get confused with the diffusion (when $\kappa^* = O(1)$). It follows that the probability for $\pi_{MPA} = \pi^*$ is always smaller than unity.
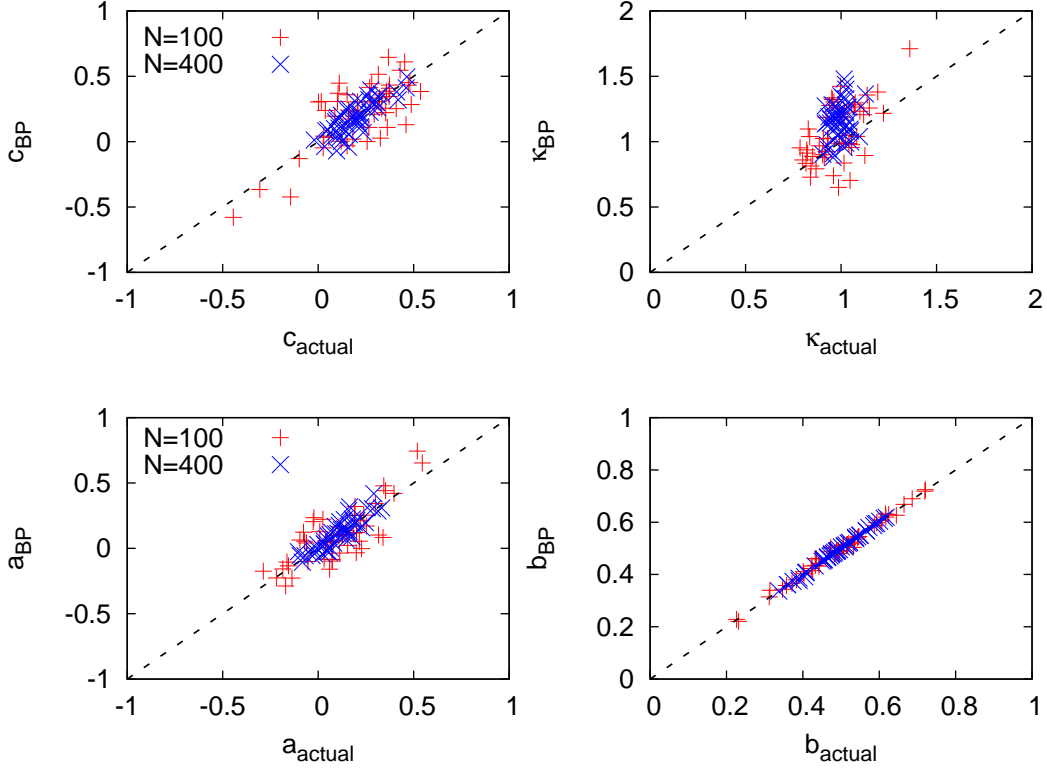
FIG. 7: Scatter plot of the parameter estimations using the BP method in the case of a 2D incompressible flow with the rate of stretching $a^* = 0.1$, the shear $b^* = 0.5$, the vorticity $c^* = 0.2$ and the diffusivity $\kappa^* = 1$. Red (blue) points refer to the case of a number of tracked particles $N = 100$ ($N = 400$) and the number of measurements is 50.

III. DISCUSSION

The message-passing algorithms discussed here were shown to ensure efficient, distributed and accurate learning of the parameters governing the stochastic map between two consecutive images recording the positions of many identical particles. The general method was illustrated in a model relevant for the tracking of particles in fluid dynamics. It was shown that parameters of the flow transporting the particles could be efficiently and reliably predicted even in situations where a strong uncertainty in the particles' trajectories is present.

We introduced and compared two techniques to approximate the likelihood that the dynamics of the particles is compatible with the displacements observed in the experimental snapshots. The first is based on finding the most probable trajectories of the particles between the times of the two images. The second corresponds to evaluating the probabilistically weighted sum over all possible trajectories. The latter is a #P-complete problem and its solution is approximated by Belief Propagation (BP), as implemented via a message-passing algorithm. BP was shown to become exact for the simplified "random-distance" model we introduced here. In general, the effect of loops in the graphical model for the tracking problem remains nonzero even in the thermodynamic limit of a large number of tracked particles. Preliminary analysis of the loop corrections to BP did not display any immediately visible structure, yet detailed analysis of this point is left for future work. Another interesting direction is the development of learning algorithms (both MCMC and message-passing) specifically designed to provide estimations of appropriate observables, e.g. the sum of the square of the distances traveled by the particles, whence the parameters of the dynamics, e.g. the diffusivity, can be estimated. This could lead to further reductions in the computational time and it will be of interest to test whether the superiority in speed we found here for BP as compared to the FPRAS Monte-Carlo scheme (see SI) still holds. The price of a single observable is that the log-likelihood curves in Fig. 4 offer more complete information, namely a systematic way to gauge the error bars on the inferred parameters.

The algorithms presented here can be carried over to the tracking of other types of "particles", e.g., those of biological interest. Motile bacteria in colonies, eukaryotic cells or fluorescent bio-molecules provide relevant examples. As it was stressed here, our methods are poised to deal with dense conditions where a strong overlap among the various particles' trajectories are present. An additional use of the techniques introduced here is to compare different

models of transport, e.g., purely diffusive, directed, active, etc. The Bethe expression given by the BP approximation can be taken as an approximation for the log-likelihood of the various models, which are then compared by standard model selection tests. The validity of the various models postulated for the dynamics of the tracked objects can thus be quantitatively compared.

Our main conclusion is that the BP method gives accurate results and its computational burden is comparable to identifying the most probable trajectories. The accuracy of BP was shown to compare extremely well with exact results and to improve rapidly as the dimensionality of the problem increases. The BP-based technique allows generalization to reconstruction of a multi-scale from particle images in two sequential snapshots. BP can also be adapted to the case where trajectories of the particles are reconstructed from their positions in several (> 2) images (see SI). In conclusion, the formulation of particle tracking as an inference problem permits tackling it systematically and introducing message-passing methods that are highly effective in diverse applications.

Appendix A: Implementation Details

Generating the data   To generate the set of positions $\boldsymbol{x}_i$ in the original image we proceed as follows. We place N particles with positions selected uniformly at random in a d-dimensional box of size $N^{1/d}$, i.e. each of the d coordinates of any of the N particles is i.i.d. on the interval $(0, N^{1/d})$. We choose the set of governing parameters, $\boldsymbol{\theta} = (\kappa, \hat{s})$, describing the diffusivity and the flow transporting the particles, in such a way that the displacements of the particles in the acquisition time of the images (chosen as the unit of time) is of order unity (we remind that the unit of length is chosen as the typical inter-particle distance). Namely, since the typical displacement will be of the order sr and $r = O(N^{1/d})$, we need that each component of the matrix $\hat{s}$ be $O(N^{-1/d})$. We therefore performed the rescaling $s = s^* N^{-1/d}$, with $s^* = O(1)$. To produce a stochastic map for the N particles from the original image to their respective positions in the successive image, we generate a d-uple of coupled Gaussian variables $\boldsymbol{r}_i$ with covariance matrix M^{-1} according to eq. (8-9) and then output $\boldsymbol{y}_i = \boldsymbol{r}_i + W\boldsymbol{x}_i$, where the evolution operator W is defined in eq. (9). Note that this setting guarantees that, for sufficiently large N, the number of particles leaving the box is much smaller than those staying in the box. For the purpose of our validation, particles which leave the box are treated as if the box did not exist.

Computing the actual estimate, $\boldsymbol{\theta}_{\text{actual}}$ Given the set of positions $\boldsymbol{x}_i$ and $\boldsymbol{y}^j$ and the actual permutation π^* (in practice, π^* can be taken as the identity without any loss of generality), we consider the likelihood eq. (1) to be a function of the set of parameters $\boldsymbol{\theta}$. The point in the $\boldsymbol{\theta}$ space where the likelihood achieves its maximum is called "the actual estimate" and it is denoted as $\boldsymbol{\theta}_{\text{actual}}$. The estimate $\boldsymbol{\theta}_{\text{actual}}$ is found using the Newton's method. Note that finite size effects make that $\boldsymbol{\theta}_{\text{actual}}$ typically differ from the respective original value, $\boldsymbol{\theta}^*$, by deviations that are $O(N^{-1/2})$.

Computing the MPA estimate, $\boldsymbol{\theta}_{\text{MPA}}$ Computation of the Most Probable Assignment (MPA) estimate is split into two parts. First, for the given set of parameters $\boldsymbol{\theta}$ we look for the permutation, π_{MPA}, which maximizes the likelihood eq. (1); second, we maximize the resulting likelihood with respect to $\boldsymbol{\theta}$ using the Newton's method. Exact algorithms are available to achieve the first task [9–11], and we utilize in our simulations the zero-temperature Belief Propagation algorithm of [11], as the latter is considerably faster than other alternatives. In the zero temperature limit, $\beta \to \infty$, the BP equations (5) become

$$\overline{h}^{i \to j} = -\max_{k \neq j}(\underline{h}^{k \to i} + \ln P_i^k), \qquad \underline{h}^{j \to i} = -\max_{k \neq i}(\overline{h}^{k \to j} + \ln P_k^j). \qquad \text{(A1)}$$

The MPA matching is reconstructed from the fixed point of eq (A1) as follows

$$\pi_1(i) = -\text{argmax}_k(\underline{h}^{k \to i} + \ln P_i^k), \qquad \pi_2(k) = -\text{argmax}_i(\overline{h}^{i \to k} + \ln P_i^k). \qquad \text{(A2)}$$

We initialize the messages $\{\overline{h}\}, \{\underline{h}\}$ at random, and solve eqs. (A1) iteratively till the consistency condition $\pi_2[\pi_1(i)] = i$ for all $i = 1, \ldots, N$ is satisfied. Note that this scheme is faster than enforcing the convergence of the BP equations (A1) to a fixed point. *A priori* we could reach a configuration of messages where the condition is satisfied and the corresponding configuration is not the best matching because the fixed point was not reached yet. However, in reality we never observed this situation when the number of particles is sufficiently large.

Computing the BP estimate, $\boldsymbol{\theta}_{\text{BP}}$ The task here is to find the minimum of the Bethe free energy eq. (6) over the set $\boldsymbol{\theta}$, where $\{\overline{h}\}, \{\underline{h}\}$ solve eq. (5) at $\beta = 1$ and given $\boldsymbol{\theta}$. Our implementation of this task is as follows. We initialize the algorithm with randomly generated $\{\overline{h}\}, \{\underline{h}\}$, solve BP equations iteratively and then minimize the resulting Bethe free energy using the Newton's method. Each time the Newton's method calls for the value of the free energy at a certain value of parameters $\boldsymbol{\theta}$, each BP message is updated on average m times and then the value of the Bethe free energy is computed. We typically used $m = 5$ or $m = 10$. In general this small number of iterations is not sufficient to reach the fixed point for each value of the parameters, however, we found empirically that reaching the fixed point at every step is unnecessary as long as the Newton's method finally converges (which was always the case in our implementation).

Mixing the BP iterations with the Newton's method updates accelerates the algorithm significantly. The running time of this mixed implementation is quadratic in the number of particles. Moreover, the following procedure allows to attain a linear scaling without much of a quality sacrifice: for each particle i, we retain only those j's such that P_i^j exceeds a predefined small constant C. As the size of the system grows, this truncation identifies only $O(1)$ number of possible partners for each particle resulting in the linear scaling for the algorithm running time.

Appendix B: Cavity solution for the random distance model

We give here a brief exposition to the cavity solution of the "random distance" model defined in the main text. More details about the method itself, applied to a different but related model of random matching, can be found in

[22–24, 31].

In the random distance model the likelihood (1) becomes

$$\mathcal{L}(\{\sigma\}|\kappa) = C\left(\{\sigma\}\right) \prod_{(i,j)} \left[P_i^j(d_i^j|\kappa)\right]^{\sigma_i^j} \left[\frac{1}{N}\right]^{1-\sigma_i^j}, \tag{B1}$$

where $P_i^j(d_i^j|\kappa) = \frac{1}{\sqrt{2\pi\kappa}}e^{-\frac{1}{2\kappa}(d_i^j)^2}$. Note that $\sum_{(i,j)}(1-\sigma_i^j) = N(N-1)$ and thus the term with $1/N$ in eq. (B1) is contributing as a multiplicative constant and can be ignored. Hence, the belief propagation equations (5) hold for the random distance model, the Bethe free energy is computed according to eq. (6), and the respective expression for the ground state energy is

$$E = \sum_{(ij)} \max(0, \overline{h}^{i\to j} + \underline{h}^{j\to i} + \ln P_i^j) - \sum_i \max_{i\in\partial j}(\underline{h}^{j\to i} + \ln P_i^j) - \sum_i \max_{j\in\partial i}(\overline{h}^{i\to j} + \ln P_i^j). \tag{B2}$$

The belief propagation solution applied to an instance of the random distance problem becomes asymptotically and statistically exact in the thermodynamic limit $N \to \infty$. This statement is equivalent to exactness of the so-called replica symmetric solution made in [22–24, 31] for a different but related random matching model. We tried to detect instabilities towards replica symmetry breaking in the random distance model, but failed to find any. This empirical evidence leads us to conjecture that one may be able to extend the rigorous local weak convergence method of [25] or independent combinatorial proof of [32] to the random distance model.

Population dynamics solution   Population dynamics [33] is a general method to solve the belief propagation equations on a infinite sample without actually generating or storing the whole sample. The assumption here is that all the quantities of interest are self-averaging, i.e., their values are asymptotically equal to respective averages over large sample. For the matching problem, a trick introduced in [24, 31] exploits the effective sparseness of the fully connected factor graph. The main observation is that the best matching has energy proportional to N whereas a random permutation has energy proportional to N^2, and only permutations with energy proportional to N contribute to the log-partition function and other quantities of interest. Edges with $O(N)$ weights almost never contribute to these configurations. The number k of edges adjacent to a variable whose weights are smaller than a certain constant c is described by the following Poissonian distribution with mean c:

$$R(k) = \binom{N}{k} \left(\frac{c}{N}\right)^k \left(1 - \frac{c}{N}\right)^{(N-k)} \to_{N\to\infty} e^{-c}\frac{c^k}{k!}. \tag{B3}$$

Additionally to these k edges we always consider an edge connecting the variable under consideration to its actual image. We thus end up with a bipartite factor graph with the connectivity degree $1+k$, where k fluctuates according to eq. (B3). As $c \to \infty$ we recover the original problem, but in practice moderate values of c are sufficient to achieve numerically the asymptotic regime.

Let us now turn to the cavity equations explaining the relations among the probabilities of the messages $\underline{h}^{i\to j}$, $\overline{h}^{i\to j}$ at the fixed point of eq. (5) on a very large graph. In fact, there are two kinds of BP messages, these describing transitions between i and its actual image and also transitions between i and nodes which do not correspond to i's actual image. Our notation for the two distributions are $\tilde{Q}(h)$ and $Q(h)$, respectively. The resulting cavity equations are

$$\tilde{Q}(h) = \sum_k R(k) \int \prod_{i=1}^k dd_i\, U(d_i) \int \prod_{i=1}^k dh_i\, Q(h_i)\, \delta(h - \mathcal{F}(\{h_i\})), \tag{B4}$$

$$Q(h) = \sum_k R(k) \int dd_0\, P(d_0) \prod_{i=1}^k dd_i\, U(d_i) \int dh_0\, \tilde{Q}(h_0) \prod_{i=1}^k dh_i\, Q(h_i)\, \delta(h - \mathcal{F}(\{h_i\}, h_0)). \tag{B5}$$

The quantity $\mathcal{F}(\{h_i\})$ satisfies eq. (5). Let us recall that $P(d) = \frac{1}{\sqrt{2\pi\kappa^*}}e^{-\frac{1}{2\kappa^*}d^2}$ is the distribution of displacements; and $U(d) = 1/c$ for $0 < d < c$, and $U(c) = 0$ otherwise.

Population dynamics is the method we utilized to solve the cavity equations (B4-B5). In this method both distributions $Q(h)$ and $\tilde{Q}(h)$ are represented by a pool of N_{pop} numbers initialized at random. In one sweep on the population dynamics we repeat N_{pop} times the following procedure:

- Iteration of eq. (B4): Draw random number k according to eq. (B3), draw k random numbers d_i uniformly at random from interval $(0, c)$, and draw k random numbers from the pool representing the distribution $Q(h)$.

Then use eq. (5) to compute a new h and substitute it on a place of a random element in the pool representing $\tilde{Q}(h)$. Note that in the case when $k = 0$ from the definition of message h, the value of h has to be large $h \to \infty$. In practice, we take $h = h_{\max}$.

- Iteration of eq. (B5): Draw random number k according to eq. (B3), draw k random numbers d_i uniformly at random from interval $(0, c)$, and draw k random numbers from the pool representing the distribution $Q(h)$. Draw one random number d from the Gaussian distribution with variance κ^*, and one random number from the pool representing $\tilde{Q}(h)$. Then use eq. (5) to compute a new h and use it to replace a random element in the pool representing $Q(h)$.

We iterate the procedure till convergence. Knowing the convergent $Q(h)$ and $\tilde{Q}(h)$ allows us to evaluate various sample averaged quantities of interest, such as the density of the free energy (6). Note that to calculate the first term in eq. (6), the term needs to be split in two parts, correspondent to the actual (ij) pairs (particle and its image) and "confused" pairs. In practice we compute each of the terms in eq. (6) N_{pop} times, each correspondent to new choice of randomness. Then, we do few (around 10 in practice) sweeps to update the pools representing $Q(h)$ and $\tilde{Q}(h)$ and repeat the measurement to eliminate sampling errors. Similar scheme also applies to evaluating average of the second derivative of the free energy.

Let us also mention for completeness that the system of cavity equations (B4-B5) allows an elegant and compact direct representation in the asymptotic $c \to \infty$ limit, in the spirit of [22, 23] where this trick was originally proposed for the random matching model. Following [22, 23] we define a generalized Laplace transform as

$$e^{-G(l)} \equiv \int_{-\infty}^{\infty} dh\, Q(h) e^{-e^{l-\beta h}}, \qquad e^{-\tilde{G}(l)} \equiv \int_{-\infty}^{\infty} dh\, \tilde{Q}(h) e^{-e^{l-\beta h}}. \tag{B6}$$

Applying this transform to eq. (B4-B5) we arrive at the following set of closed equations for $G(l)$, $\tilde{G}(l)$:

$$\tilde{G}(l) = \int_{-\infty}^{\infty} dy\, e^{-G(y)} \sum_{p=1}^{\infty} \frac{(-1)^{(p-1)}}{p!(p-1)!} e^{p(l+y)} g_p(\beta), \tag{B7}$$

$$G(l) = \tilde{G}(l) - \log\left[1 - \int_{-\infty}^{\infty} dy\, e^{-\tilde{G}(y)} \sum_{p=1}^{\infty} \frac{(-1)^{p-1}}{p!(p-1)!} e^{p(l+y)} \tilde{q}_p(\beta)\right], \tag{B8}$$

where

$$g_p(\beta) = \sqrt{\frac{\pi}{2\beta p}}, \qquad \tilde{q}_p(\beta) = \sqrt{\frac{1}{\kappa^* \beta p + 1}}. \tag{B9}$$

Note, that eq. (B7) with $\tilde{G} = G$ was originally derived in [22, 23] using the replica method for the random matching problem, while the log-correction on the right hand side of eq. (B8) is specific to our problem, representing bias brought into the problem by the existence of the special permutation π^*.

Results Let us briefly discuss the results obtained by the population dynamics numerical evaluation of the cavity equations eq. (B4-B5).

First, we compare the most probable matching with the actual matching and analyze their dependence on the diffusivity κ^*. We define the Hamming distance between the actual and the most probable matchings as the fraction of edges which are present in the actual matching but are missed in the most probable matching. The comparison is shown in Fig. 8, where we plot the Hamming distance as a function of the diffusivity. The inset of the Figure shows a transition observed at $\kappa^* < 0.174(4)$ from the lower diffusivity phase, where no difference between the most probable and actual matching were detected, to the high diffusivity phase where the most probable and actual matchings are distinctly different. The Hamming distance saturates to 1 as $\kappa^* \to \infty$. Note that this phase transition is a property specific to the random distance model and that it was not observed in our particle simulations in $2d$ and $3d$ discussed in the main text.

To estimate the most probable value of the diffusion constant κ based on the observed data (set of mutual distances) we need to maximize the full likelihood $Z(\kappa)$, eq. (2), with respect to κ, i.e. to minimize the free energy of the model. In the limit, $N \to \infty$, the minimum of the free energy within the random distance model is always achieved at $\kappa = \kappa^*$. However, the minimum gets flatter as the the actual diffusivity κ^* grows. In a finite size system the statistical error in estimating κ^* is $1/\sqrt{(N\mathcal{F}'')}$. Let us remind here that if the actual matching is known then the statistical error is $\kappa^* \sqrt{(2/N)}$. In Fig. 8, we plot (in log-log-scale) the second derivative of the free energy as a function of κ^*. We observe that for $\kappa^* < 1$ the second derivative is roughly $\mathcal{F}'' = 1/[2(\kappa^*)^2]$, hence in this regime the statistical error
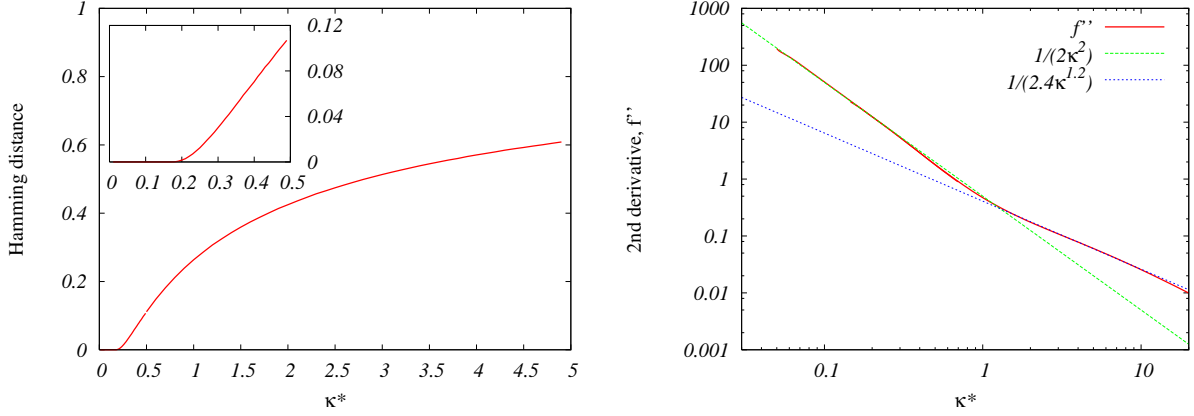
FIG. 8: Left: Result of the zero temperature population dynamics with $N_{\text{pop}} = 10000$, $c = 20$, $t_{\text{max}} = 10000$. Distance between the actual and the most probable matchings shown as a function of diffusivity κ^*. Inset: Zoom into the region of the phase transition at low diffusivity. At about $\kappa_c \approx 0.174(4)$ the most probable matching ceases to be equal (asymptotically) to the actual matching. Right: The second derivative of the free energy at κ^* as a function of κ^*. We observe the change of the slope at $\kappa^* \approx 1$ from $1/[2(\kappa^*)^2]$ to a larger value.

is of a similar origin as if the actual matching would be known. For $\kappa^* > 1$, the scaling of the second derivative changes to roughly proportional to $1/\kappa^*$, hence in this regime estimation of the actual κ^* from the full likelihood is more accurate. It is noteworthy to emphasize that the crossover in the second derivative (and thus accuracy of the log-likelihood based prediction) is observed at $\kappa^* \approx 1$, whereas the phase transition in the Hamming distance (between the actual and most probable matchings) takes place at the much lower values $\kappa^* = \kappa_c \approx 0.174(4)$.

Appendix C: Comparative analysis of BP and MCMC algorithms

In the body of the paper we stated the problem of learning the parameters of the flow/diffusion from particle tracking data in terms of the partition function of an associated matching problem. One important point articulated in the manuscript is that, even in the regime where the most probable matching is very different from the actual one, computing the partition function enables us to estimate the parameters accurately. Note, that to the best of our knowledge all the methods used so far in particle tracking reconstruction/learning, have been relying solely on a single matching.

Our results show that BP is both efficient and accurate for approximating the maximum of the partition function. We have used an MCMC-learning algorithm, but our original focus was primarily on assessing the quality of the BP by comparing it to the MCMC near-to-exact result. To guarantee near-to-exact result from the MCMC algorithm the running time is impractical. In this Section we give some additional details and comment also on comparative speed performance of the two learning algorithms.

MCMC sampling is the most common method for estimating properties of the Boltzmann distribution. However, standard MCMC approaches sample from the given distribution aiming to evaluate certain expectation values and they are not directly suitable for accurate counting (evaluation of the partition function) required for our purposes. A considerable speed up in sampling of lower cost configurations is usually achieved via Metropolis-Hastings implementation of the MCMC, however, in order to compute the partition function (or the free energy) one has to numerically integrate the energy function over a range of temperatures, and this integration is computationally costly.

To resolve this problem we have used a special MCMC algorithm uniquely designed for this purpose – a somewhat simplified version of the Fully Polynomial Randomized Approximation Scheme (FPRAS), originally introduced for computing permanents (and this is what our partition function is) in [17]. The unabridged version of this FPRAS algorithm is guaranteed to give a value that is not worse than $(1 + \epsilon)L$ (L being the exact value) in $O(N^{11})$ and poly$(1/\epsilon)$ computational time [17]. A description of our simplified algorithm follows.

Let us denote the matrix whose permanent we want to compute by A. In our particular case, A contains the likelihoods of pairs of particles in different time steps corresponding to each other. The algorithm works in stages. It starts with a constant matrix A', whose elements are all equal to the largest value in A, a_{max}, and whose permanent is easily computed as $(a_{max})^n n!$. In each stage i, elements of A' are reduced by a constant factor to $a'_{uv} = \max\{\exp(-1/2)a'_{uv}, a_{uv}\}$, thus bringing A' closer to A. Using S (a parameter) samples from all possible

perfect matchings (i.e., permutations), an approximate ratio r_i is computed of the new value of the permanent of A' and its old value (before reducing elements of A' in step i). The samples are obtained using a Metropolis-Hastings MCMC, where neighbors of a state are all matchings with any two pairings swapped, and the probability of accepting a proposed step is the ratio between products of values of A' corresponding to new and old matchings. This MCMC is run for T (a second parameter) steps for each sample. Finally, the algorithm finishes when the matrix A' becomes sufficiently close to the original A. The approximate value of the permanent is then obtained as $\prod_i r_i \cdot (a_{max})^n n!$. The two parameters, S and T, determine the time complexity of the algorithm, which is $O(ST)$.
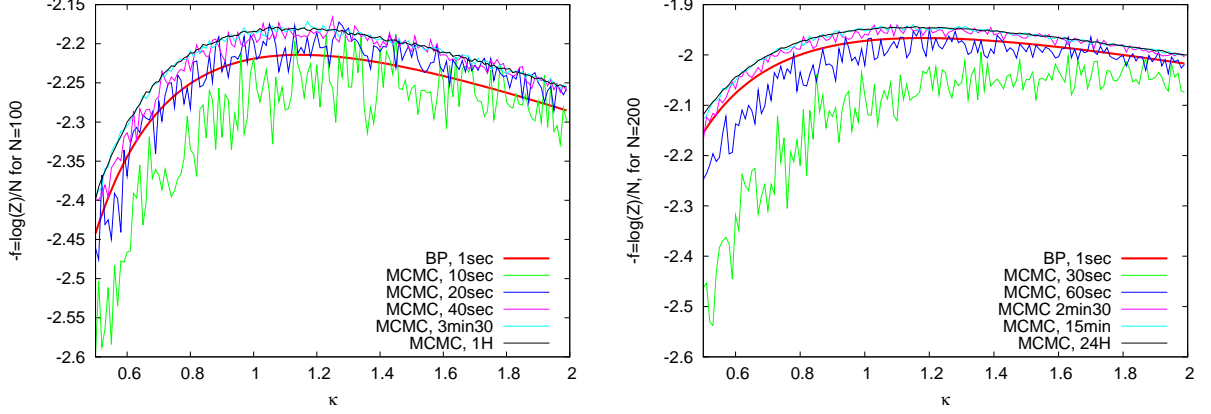

FIG. 9: Logarithm of the partition function computed from the belief propagation algorithm (in red thick line) and from the Monte Carlo described in the text. The Monte Carlo is guaranteed to find an ϵ approximation of the exact value in $O(N^{11})$ time. We, however, run it for a much shorter periods and analyze how does its performance depend on the running time. We find that in order to compare with BP in the estimation of the maximum, MCMC needs simulations that are at least one order of magnitude longer.

To facilitate comparison of the MCMC and BP efficiency (speed), we downgraded strict quality requirement of the original FPRAS-MCMC and conducted simulations to access its deterioration in quality with the computational time (and number of samples) decrease. For this comparison we have used instances of the particle tracking described in Section A. The results, shown in Fig. 9, lead to conclusion that for comparable performance we need about an order of magnitude longer running time for the MCMC. Note, however, that we tested only specific implementation of the MCMC algorithm, and obviously further comparative analysis of BP-based, MCMC-based and possibly other learning schemes will be necessary.

We find it useful to complete this Section by discussing another important point related to the comparison of the MCMC and BP efficiency for evaluating the maximum of the partition function. This is a crucial quantity for learning as it provides the most likely estimate of the flow/diffusion parameters. We have observed that using BP provides an explicit estimate not only for Z, but also for its first and second derivatives over the parameters. This observation has helped us to significantly accelerate the BP-based learning thus streamlining the search for the maximum of the partition function over the parameters. The acceleration is achieved via the use of Newton method in parallel with BP iterations. The trick itself gives an advantage to BP and it is currently not clear if similar accelerations are possible to achieve in the MCMC-based learning.

Appendix D: Possible extensions

In this section we discuss possible extensions of our approach that will be interesting to pursue in future work. We nevertheless find it useful to briefly discuss them here, in order to give a better idea about possibilities (and limitations) of a systematic probabilistic approach to particle tracking.

Incorporating uncertainty   Our approach allows simple modifications to incorporate various types of uncertainty. For example, one can account for particles leaving and entering the box (focal volume in the experimental realization) via the following relaxation of the constraint $C(\{\sigma\})$ in eq. (1),

$$C(\{\sigma\}) \equiv \prod_j \left[\delta\left(\sum_i \sigma_i^j, 1 \right) + e^{-\mu} \delta\left(\sum_i \sigma_i^j, 0 \right) \right] \prod_i \left[\delta\left(\sum_j \sigma_i^j, 1 \right) + e^{-\mu} \delta\left(\sum_j \sigma_i^j, 0 \right) \right], \quad \text{(D1)}$$

where the parameter μ is interpreted as a chemical potential that accounts for the fluctuations in the number of particles. The BP equations can be rewritten for this form of likelihood and the penalty parameter μ would have to be varied so as to optimize the likelihood and allow appropriate number of non-matching particles.

Processing of multiple snapshots Our approach can also be adopted to inference and learning based on more than two subsequent snapshots. If the incremental displacement of particles between two subsequent snapshots depends only on particle positions at the early snapshot in the pair and does not depend on other details of the preceding evolution of the particles, then generalization of our approach consists in treating matchings between each pair of snapshot in the sequence independently, stating the total likelihood of the sequence as a product of the respective expressions eq. (1) for the pairs, and making global optimization over the governing parameters $\boldsymbol{\theta}$ for the entire product.

If the motion of particles has memory spanning over m snapshots, then our approach can still be implemented, however this will require knowledge or estimation for the joint probability of observations $P(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m | \boldsymbol{\theta})$. Note that in this correlated case already the subtask of inferring MPA assignment, equivalent to the so-called multi-matching problem [24, 31], becomes significantly more difficult than in the non-correlated case.

Some sort of predictor-corrector scheme could also be investigated to solve the case with memory, or to achieve a speed-up of multiple snapshot processing. These promising possibilities are yet to be investigated and tested.

Use of marginal probabilities We concentrated on inference of flow parameters in regime where tracking of individual particles is no longer possible. Note, however, that once the correct parameters are estimated the belief propagation provides also estimates of probabilities with which a given particle from the first snapshot moved to a given position in the second snapshot. This information may be useful in cases where we are interested in information about the trajectories of individual particles. The values of these probabilities also provide concrete information about the level of uncertainty where a given particle moved.

Note, however, that if a particle is matched to the position of its most probable image, the resulting configuration may even not be a one-to-one mapping. Indeed, a position in the second image might be the most probable one for multiple particles from the first image. Decimation-like techniques, of the type used to solve similar inconsistency problems in the constraint satisfaction problems [34], can then be used to obtain a consistent one-to-one mapping. In either case, even though the resulting configuration may not be equal to the actual displacement, it may still provide a useful visual information.

Learning multi-scale flow In this paragraph we describe how the learning framework, illustrated in the body of the manuscript on the model case of the Batchelor (single-scale) velocity, can be extended to the more general case of a multi-scale flow. The Batchelor model of velocity considered in the main text assumed that $\hat{s}$, the matrix of velocity gradients, does not depend on the particle index, i.e. in other words, all the particles in the cloud sense the same velocity gradient. In a more realistic turbulence setting the velocity gradient varies on the spatial scale correspondent to the viscous (Kolmogorov scale). Thus, when particles are seeded in a cloud exceeding in size the viscous scale, one needs to use a richer higher-parametric model, for example stated in terms of the following harmonic expansion of the instantaneous velocity field: $\boldsymbol{u}(\boldsymbol{r}) = \sum_{\boldsymbol{q}} \exp(i\boldsymbol{q}\boldsymbol{r})\boldsymbol{u}_{\boldsymbol{q}}$, where the number of harmonics (number of terms in the sum) is N_q. Then, velocity advecting particle i, where labeling is according to the first of the two consecutive images, is modeled in terms of the following gradient flow, $\boldsymbol{v}_i(\boldsymbol{r}_i | \boldsymbol{r}_i^*) = \hat{s}_i(\boldsymbol{x}_i)(\boldsymbol{r}_i - \boldsymbol{r}_i^*) + \boldsymbol{v}_i^{(0)}(\boldsymbol{r}_i^*)$, where $\boldsymbol{v}_i^{(0)}(\boldsymbol{r}_i^*) = \boldsymbol{u}(\boldsymbol{r}_i^*) = \sum_{\boldsymbol{q}} \exp(i\boldsymbol{q}\boldsymbol{r}_i^*)\boldsymbol{u}_{\boldsymbol{q}}$, $\hat{s}_i^{(0);\alpha\beta}(\boldsymbol{r}_i*) = \nabla^\alpha u^\beta(\boldsymbol{r}_i) = \sum_{\boldsymbol{q}} \exp(i\boldsymbol{q}\boldsymbol{r}_i^*)iq^\alpha u_{\boldsymbol{q}}^\beta$, and $\boldsymbol{r}_i^*$ is a pre-selected and time-independent (frozen) grid point closest to the particle's initial position (in the first image), $\boldsymbol{x}_i = \boldsymbol{r}_i(t=0)$. (One possible choice for $\boldsymbol{r}_i^*$ might be $\boldsymbol{x}_i$ itself.) The stochastic dynamic equation for trajectory of particle i becomes

$$\dot{\boldsymbol{r}}_i = \boldsymbol{v}_i^{(0)}(\boldsymbol{r}_i^*) + \hat{s}_i^{(0);\alpha\beta}(\boldsymbol{r}_i^*)(\boldsymbol{r}_i - \boldsymbol{r}_i^*) + \boldsymbol{\xi}_i(t). \tag{D2}$$

These equations can be integrated over time, thus resulting in the following generalization of Eqs. (7-9) from the main text

$$P_i^j(\boldsymbol{x}_i, \boldsymbol{y}^j) = (\det M_i)^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}\tilde{\boldsymbol{r}}^\alpha (M^{-1})_i^{\alpha\beta} \tilde{\boldsymbol{r}}^\beta\right); \tag{D3}$$

$$\tilde{\boldsymbol{r}} = \boldsymbol{y}^j - r_i^* - W_i(\int_0^\Delta dt' W_i^{-1}(t)\boldsymbol{v}_i^{(0)} + \boldsymbol{x}_i - \boldsymbol{r}_i^*); \tag{D4}$$

$$M_i = \kappa\, W_i(\Delta) \left[\int_0^\Delta W_i^{-1}(t)W_i^{-1,T}(t)\,\mathrm{d}t\right] W_i^T(\Delta), \tag{D5}$$

where $W_i(t) = \exp(t\,\hat{s}_i)$. The task of reconstruction becomes to infer N_q harmonics given positions of N particles in the two snapshots, for example assuming that the diffusion coefficient κ is known. Naturally, a reliable multi-parametric reconstruction is feasible if $N_q \ll N$. Extension of our BP-based scheme to this multi-parametric setting is straightforward, as the graphical model employed to solve the problem is identical to the one described in the main text for the Batchelor model case. The main technical difficulty in learning the multi-parametric velocity will in fact

be in maximizing the partition function of the model over a larger (than in the Batchelor case) number of the degrees of freedoms (harmonics). However, and as discussed above in Section C, this problem can be dealt with efficiently by alternating Newton (parameter adjustment) steps with BP iterative steps. The results and performance of the multi-scale inference are, however, yet to be tested.

Interacting particles Let us note that extending our approach to systems of interacting particles may be possible, but it would constitute a more significant challenge, as in this case possible probabilistic models of particles' evolution are much harder to evaluate.

[1] Ballerini M *et al.* (2008) Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proc. Natl. Acad. Sci. USA* **105** : 1232–1237.

[2] Ballerini M *et al.* (2008) Empirical investigation of starling flocks: a benchmark study in collective animal behaviour. *Animal Behaviour* **76** : 201–215.

[3] Keller PJ, Schmidt AD, Wittbrodt J, E.H.K. Stelzer (2008) Reconstruction of Zebrafish Early Embryonic Development by Scanned Light Sheet Microscopy. *Science* **322** : 1065–1069.

[4] Saxton MJ (2008) Single-particle tracking: connecting the dots. *Nature Methods* **5** : 671–672.

[5] La Porta A, Voth GA, Crawford AM, Alexander J, Bodenschatz E (2001) Fluid particle accelerations in fully developed turbulence. *Nature* **409** : 1017–1019.

[6] Mordant N, Metz P, Michel O, Pinton JF (2001) Measurement of Lagrangian velocity in fully developed turbulence. *Phys. Rev. Lett.* **87** : 214501.

[7] Adrian RJ (2005) Particle-imaging techniques from experimental fluid mechanics. *Annual Review of Fluid Mechanics* **23** : 1–261.

[8] Masson JB *et al.* (2009) Inferring maps of forces inside cell membrane microdomains. *Phys. Rev. Lett.* **102** : 048103.

[9] Kuhn HW (1955) The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, **2** : 83-97.

[10] Bertsekas DP (1992) Auction algorithms for network flow problems: A tutorial introduction. *Comput. Optimiz. Applic.* **1** : 7–66.

[11] Bayati M, Shah D, Sharma M (2008) Max-product for maximum weight matching: convergence, correctness and LP duality. *IEEE Transactions on Information Theory* **54** : 1241–1251.

[12] Ouelette NT, Xu H, Bodenschatz E (2006) A quantitative study of three-dimensional Lagrangian Tracking Algorithm. *Experiments in Fluids* **40** : 301.

[13] Valiant LG (1979) The Complexity of Computing the Permanent. *Theoretical Computer Science* **8** : 189–201.

[14] Pearl J (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. *Morgan Kaufmann Ed.*.

[15] MacKay DJC (2003) Information Theory, Inference, and Learning Algorithms, *Cambridge University Press.*

[16] Yedidia JS, Freeman WT, Weiss Y (2005) Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory* **51** : 2282–2312.

[17] Jerrum M, Sinclair A, Vigoda E (2004) A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM* **51** : 671–697.

[18] Chertkov M (2008) Exactness of belief propagation for some graphical models with loops. *Journal of Statistical Mechanics* P10016.

[19] Gallager RG (1963) Low Density Parity Check Codes. *MIT press, Cambridge, MA.*

[20] Mezard M, Montanari A (2009) Information, Physics and Computation.*Oxford Univ. Press.*

[21] Falkovich G, Gawedzki K, Vergassola M (2001) Particles and fields in fluid turbulence. *Rev. Mod. Phys.* **73** : 913–975.

[22] Mezard M, Parisi G (1985) Replicas and optimization. *J. de Physique* **46** : L771-778.

[23] Mezard M, Parisi G (1986) Mean-field equations for the matching and the traveling salesman problems. *Europhysics Letters* **2** : 913–918.

[24] Martin OC, Mezard M, Rivoire O (2004) Frozen glass phase in the multi-index matching problem. *Phys. Rev. Lett.* **93** : 217205.

[25] Aldous DJ (2001) The $\zeta(2)$ limit in the random assignment problem. *Rand. Struct. Algo.* **18** : 381–418.

[26] Ouellette NT, Xu HT, Bodenschatz E (2009) Bulk turbulence in dilute polymer solutions. *Journ. of Fluid Mechanics* **629** : 375–385.

[27] Magde D, Webb WW, Elson E (1972) Thermodynamic fluctuations in a reacting system - Measurement by fluorescence correlation spectroscopy. *Phys Rev Lett* **29** : 705–708.

[28] Elson EL, Magde D (1974) Fluorescence correlation spectroscopy. 1. Conceptual basis and theory. *Biopolymers* **13** : 1–27.

[29] Magde D, Elson EL, Webb WW (1974) Fluorescence correlation spectroscopy. 2. Experimental realization. *Biopolymers* **13** : 29–61.

[30] Rigler R, Mets U, Widengren J, Kask P (1993) Fluorescence correlation spectroscopy with high count rates and low background : analysis of translational diffusion. *Eur. Biophys. J.* **22** : 169–75.

[31] O. C. Martin and M. Mézard and O. Rivoire (2005), Random multi-index matching problems, J. Stat. Mech., P09006

[32] S. Linusson and J. Wastlund (2004), A proof of Parisi's conjecture on the random assignment problem, Probability Theory and Related Fields, **128**, 419-440.

[33] M. Mézard and G. Parisi (2001), The Bethe lattice spin glass revisited, Eur. Phys. J. B **20**, 217.

[34] M. Mézard, G. Parisi, and R. Zecchina. Analytic and algorithmic solution of random satisfiability problems. *Science*, 297:812–815, 2002.

[35] We assumed that the velocity gradients do not change significantly between two images. $W(t)$, which is generally a time-ordered exponential, is then an ordinary matrix exponential. This simplifying assumption can be relaxed, thus allowing extension of the technique to acquisition times comparable to the viscous scale of turbulence.